

A Light for Science



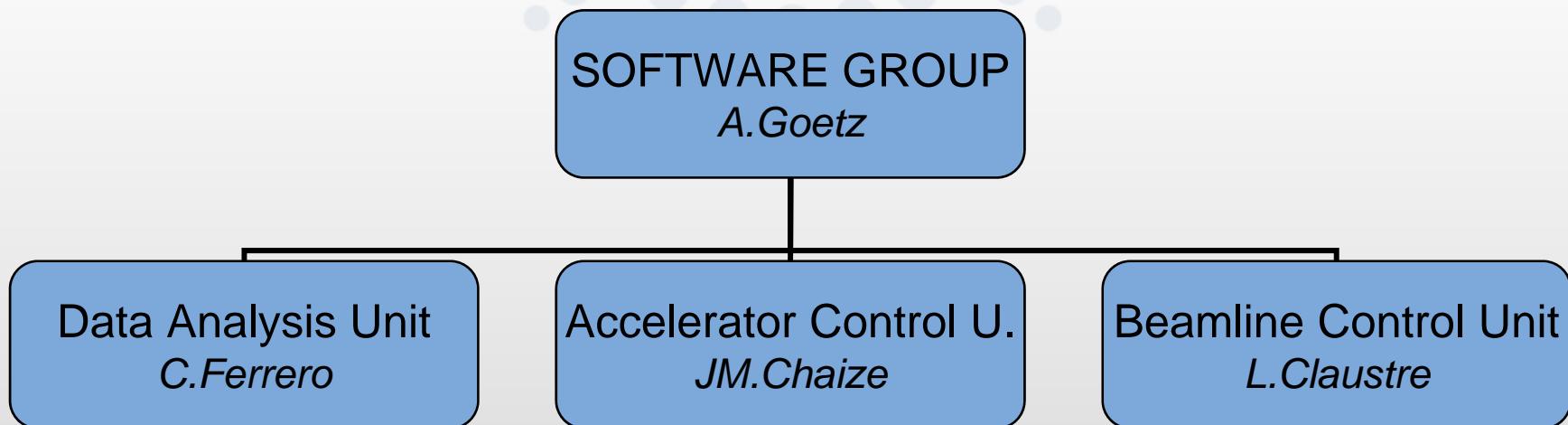
European Synchrotron Radiation Facility

# **Status and future of beamline control software at ESRF**

**E.Papillon,  
Beamline Control Unit**

# ISDD

## Instrument Services and Development Divisions



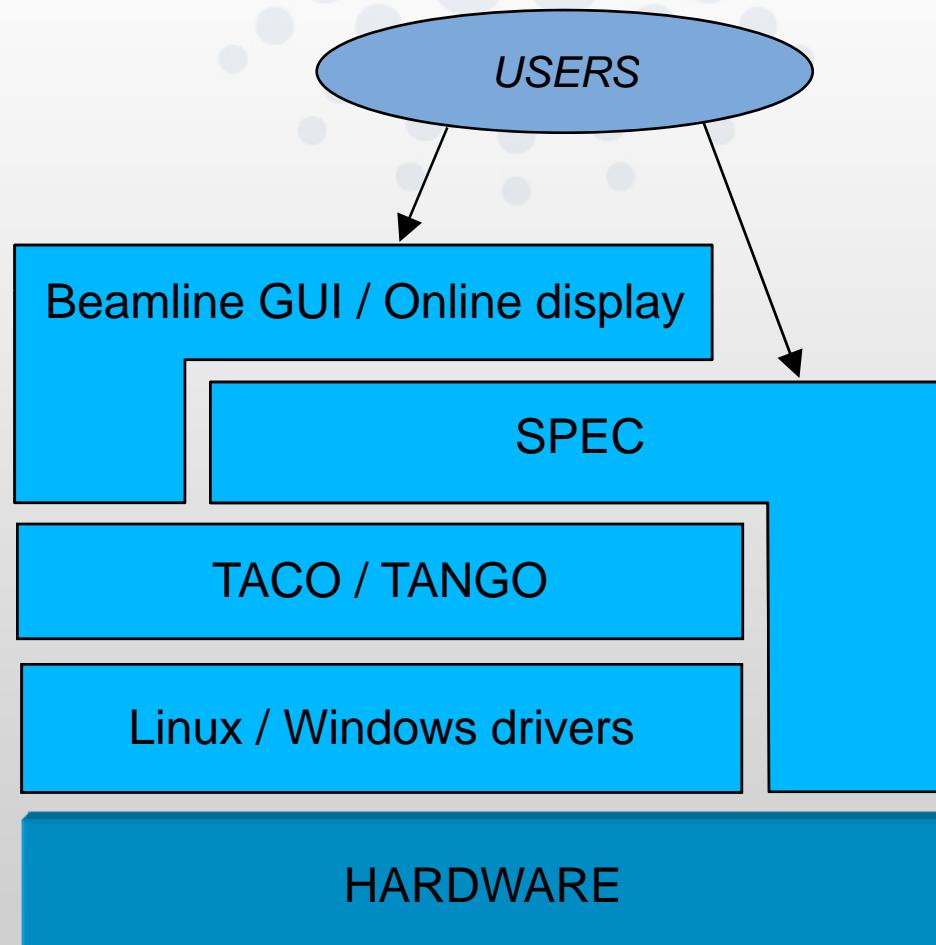
### Beamline Control Unit (former BLISS)

Support and Development of beamline control software  
29 ESRF and 12 CRG beamlines, 6 Laboratories  
16 people

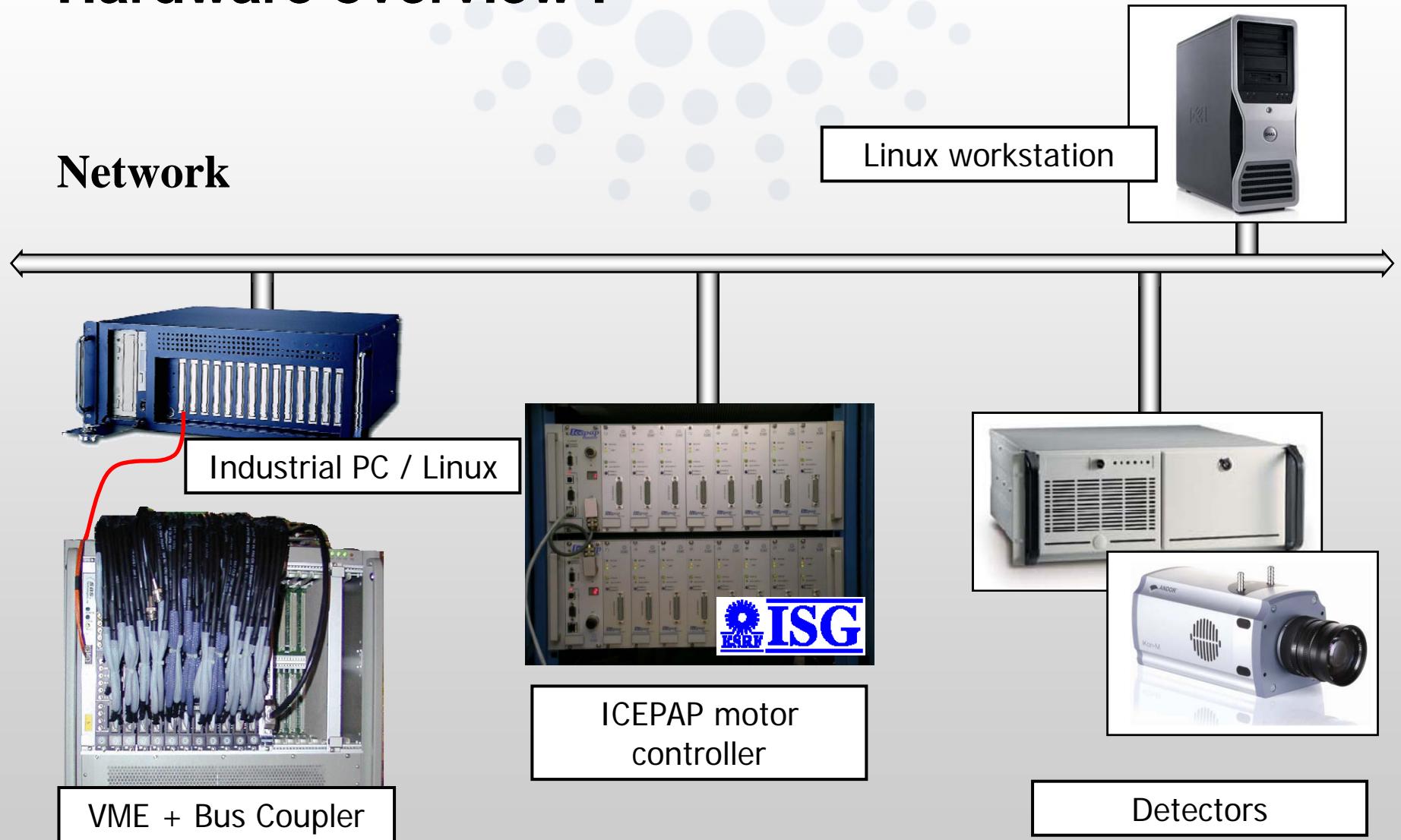
# Outline

- Beamline control today:
  - Software architecture
  - Continuous scans
  - Fast 2D detectors
  - Graphical Interfaces
- Evolution
- Conclusion

# Software overview



# Hardware overview :



# TACO / TANGO

## Control of distributed hardware

### TACO

- Developed at ESRF
- Obj. C / RPC
- Client and server API in C/python/matlab/labview/...
- Only commands

*Widely used on beamlines:*

~100 different servers  
7000+ devices exported

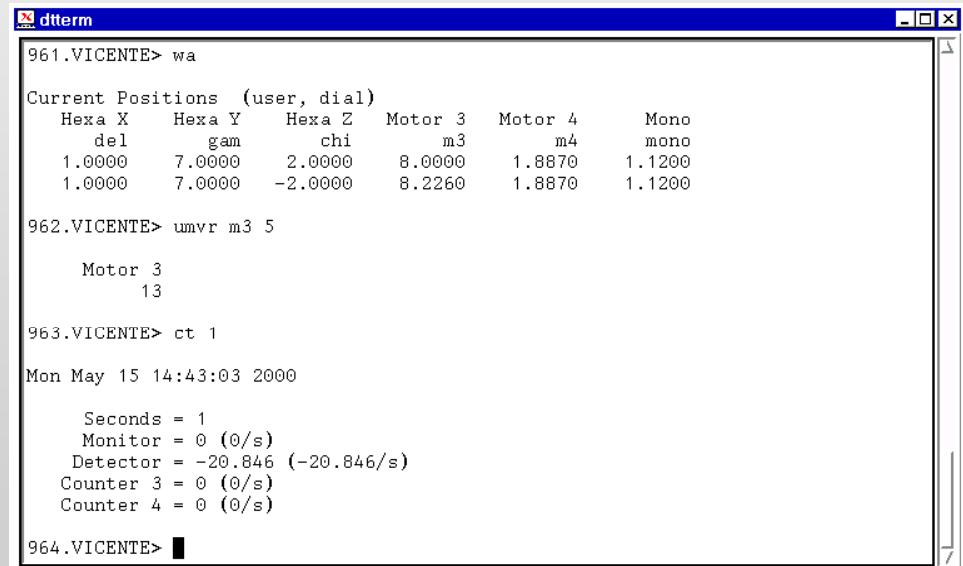
### TANGO

- Collaboration ESRF, Elettra, Soleil, ALBA, DESY
- C++ / CORBA
- Client and server API in C/C++/python/java
- Commands, attributes, properties
- Multi-thread
- Events support

*All new servers in tango*

# What is SPEC for the beamline ?

- Command Line Interface
- Main sequencer
- Handle diffractometers geometries
- Device controller:
  - *Integrated in SPEC*
  - *Taco interface:*
    - Motors / Counters
    - MCA / CCD
- Generic Interface to:
  - Serial Line
  - GPIB
  - Socket
  - Taco / Tango



```
dterm
961.VICENTE> wa
Current Positions (user, dial)
   Hexa X    Hexa Y    Hexa Z    Motor 3    Motor 4    Mono
      del      gam      chi      m3      m4      mono
   1.0000    7.0000   2.0000    8.0000   1.8870   1.1200
   1.0000    7.0000  -2.0000   8.2260   1.8870   1.1200

962.VICENTE> umvr m3 5
          Motor 3
          13

963.VICENTE> ct 1
Mon May 15 14:43:03 2000
          Seconds = 1
          Monitor = 0 (0/s)
          Detector = -20.846 (-20.846/s)
          Counter 3 = 0 (0/s)
          Counter 4 = 0 (0/s)

964.VICENTE> █
```

# SPEC macros

- **Macro motors:**

- *Physical* : ICEPAP (socket), piezo (GPIB), ...
- *Calculational* : slit offset/gap, 3 legs table, ...

- **Macro counters:**

- *Physical* : eurotherm, lakeshore, ...
- *Calculational* : ratio, attenuators transmission, ...

- **Other Macros:**

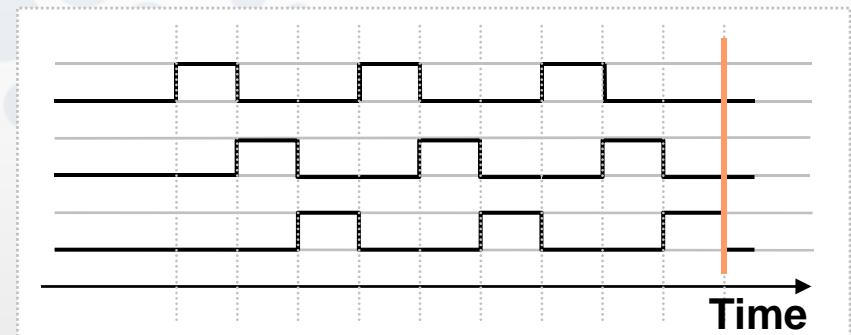
- Other type of devices : beam shutters, filter box, ...
- New sequences : exafs scans, gap scans, ...

→ Very flexible : many devices/sequences written  
→ limited data types, no debugger, no multi-threads

# Continuous scans

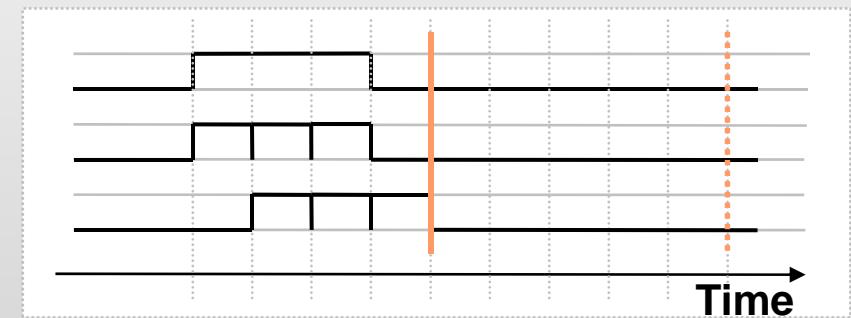
- *From Sequential*

MOVE  
EXPOSURE  
READOUT



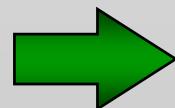
- *To Parallel*

MOVE  
EXPOSURE  
READOUT



- *Practical Issues*

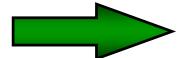
- Distributed System
- Multiple detectors
- Short exposure time



- Synchronization
- Data Buffering

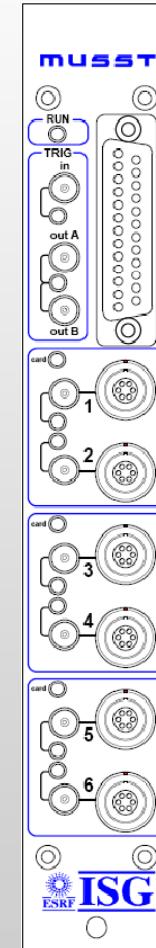
# Continuous scans : synchronization

- *By software :*
  - Synchro. with time: no wiring, good on single host
- *By hardware :*
  - Needs wiring, good for distributed hosts
  - Synchro. with time : **P201, OPIOM** boards
  - Need more : motor position, mixed time/position, ...



**MUSST board**

- **Very flexible event generator**
- **6 x input signals :**
  - Digital counter
  - Analog
  - Positioning (motor steps and encoders)
- **16 x digital I/O**
- **Host Interface : GPIB**



# Continuous scans : data buffering

- Software implementation : **HOOK**
  - Kernel driver + TACO device server
  - Hardware readout + buffering
  - For In-house developed hardware
    - PCI and VME boards
  - Software event generator
- Hardware implementation : **MUSST**
  - Encoders, motor steps
  - Counters, ADC
  - Canberra MCA (ICB)
- Data handling:
  - SPEC polls data buffers
  - Counters / motors saved by SPEC



# Continuous scans : detectors

- Canberra MCA (ICB)
  - Data buffering on MUSST
- XIA XMAP
  - PXI board – 4 channels
  - PXI/PCI bus coupler
  - External sync. Signal
  - 4MB memory / board
  - Data buffering in TACO

▶ **10 ms / spectra**

▶ **1 ms / spectra  
(one board)**



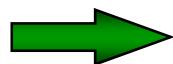
Spectras saved by device servers

Integrated ROIs transferred to SPEC (online plotting)

➔ Configuration issues  
➔ No generic counter interface

## 2D detectors

- TACO CCD interface in SPEC
  - Many CCDs supported:
    - Princeton, Sensicam, PCO, Andor, Photonic Science, ...
    - Standard macro set
    - Beam parameters, Multi-frame buffer, Parallel saving, ...
    - ✗ Per device development



### LIMA: Library for Image Acquisition

- Reuse of common code ⇒ generic procedures + interfaces
- Support recent fast detectors

## 2D detectors: LIMA

- Hardware layer
- Control layer
  - Software “features” fallback if hardware has limited capabilities:
    - ROI, Binning, Frame Accumulation
    - Basic processing and data reduction:
      - Beam parameters, ROI statistics, parallel saving, dark subtraction...
      - Specific data reduction algorithms through “plug-ins” (C++)
  - Low level: C++ (multi-threaded)
  - High level: C++ / Python (SIP)
  
  - Generic TANGO interface + detector configuration
  - Currently integrated: Pilatus, Frelon, Maxipix

## 2D detectors : ESPIA card

- Designed to interface the FReLoN 2k
- Collaboration  $\Rightarrow$  ESRF + SECAD, S.A.
- 2 Gbps fiber optic link ( $> 100$  m)
- PCI 64 bits / 66 MHz
- $\sim 180 \text{ MB/s}$  maximum data rate
- FOCLa : 2xCameraLink connectors



### FReLoN HD (PSB-2)

2048 x 1024 @ 32 fps  $\Rightarrow \sim 125 \text{ MB/s}$



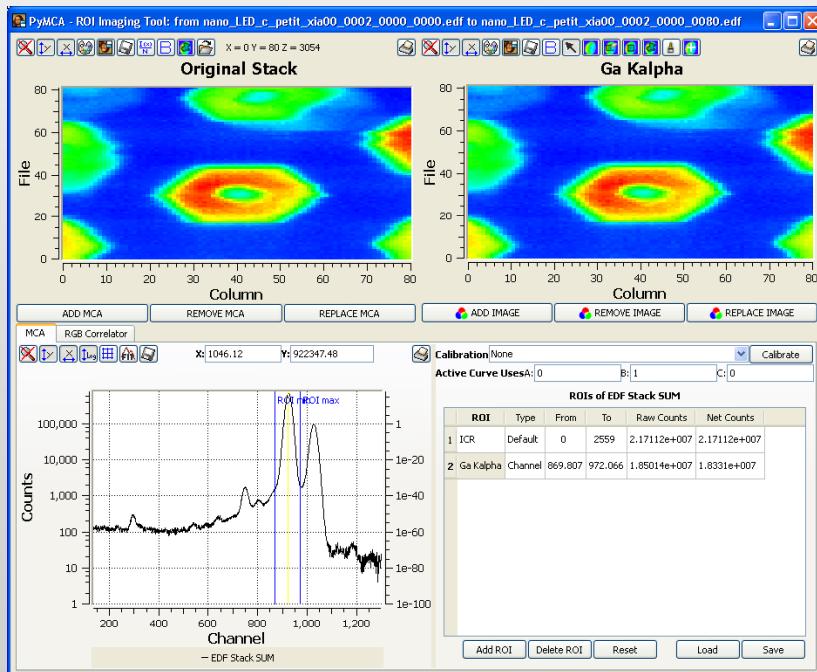
## 2D detectors : Medipix2 / Maxipix

- Pixel detector  $\Rightarrow$  photon counting
- $256 \times 256 - 13$  bit
- 0.3 ms readout time
- Max: **1400 fps  $\Rightarrow$  180 MB/s**
- Maxipix: Medipix2 array  $5 \times 1,2 \times 2$
- Image reconstruction in LIMA  
**→ 1000 fps  $\Rightarrow$  650 MB/s**
- ESPIA next generation:
  - ✓ PCI-Express 8x lanes
  - ✓ 4 fiber optic links @ 250 MB/s  $\Rightarrow$  1 GB/s

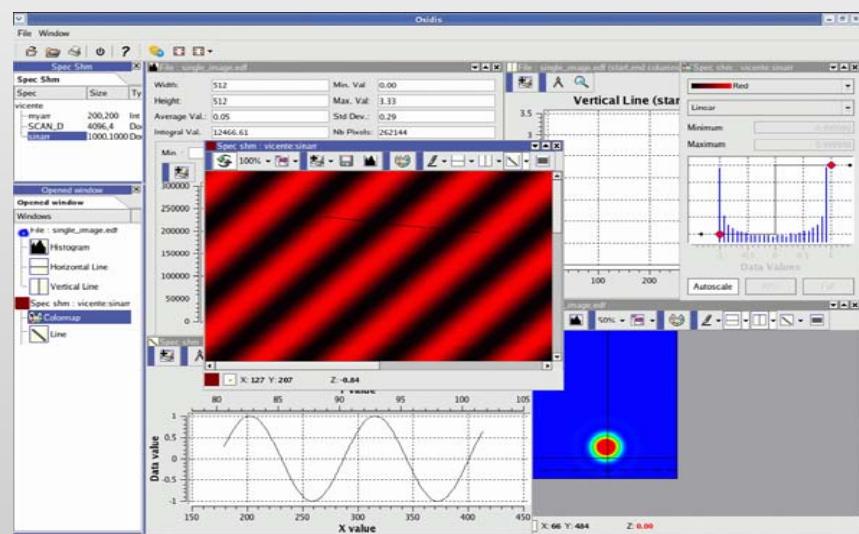


# Online Display

- Standalone tool for visualisation (python/Qt)
- Work online : use **SPEC** shared memory
- Work offline : specfiles, edf files, hdf files



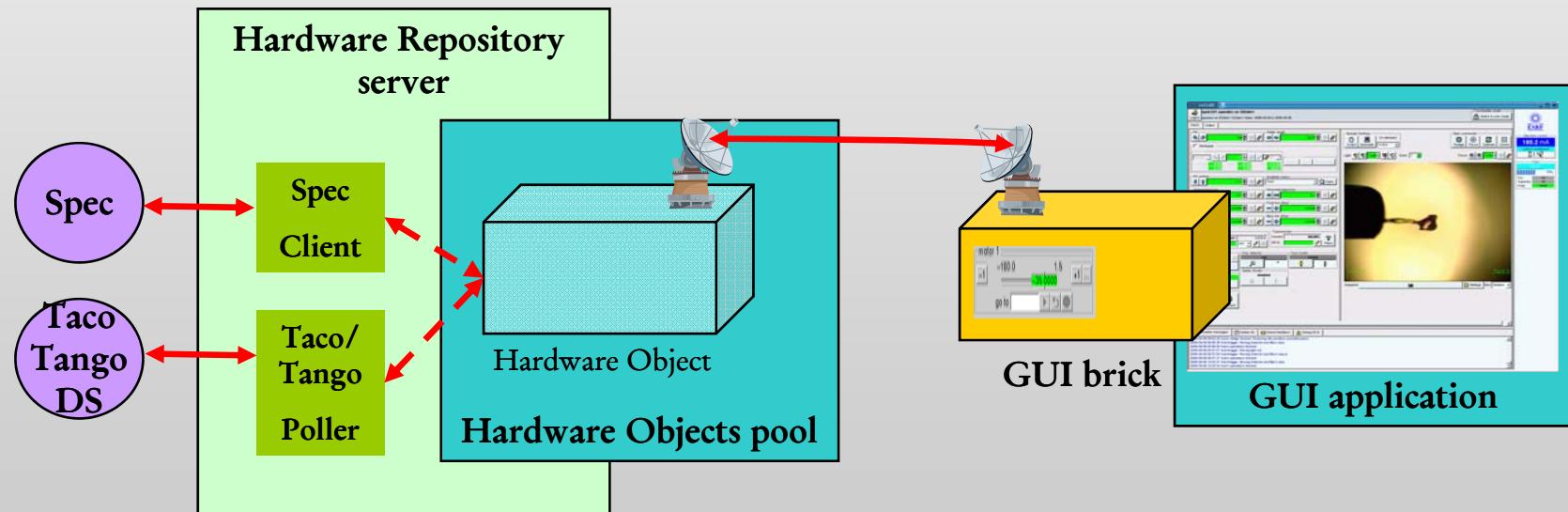
**PyMCA**



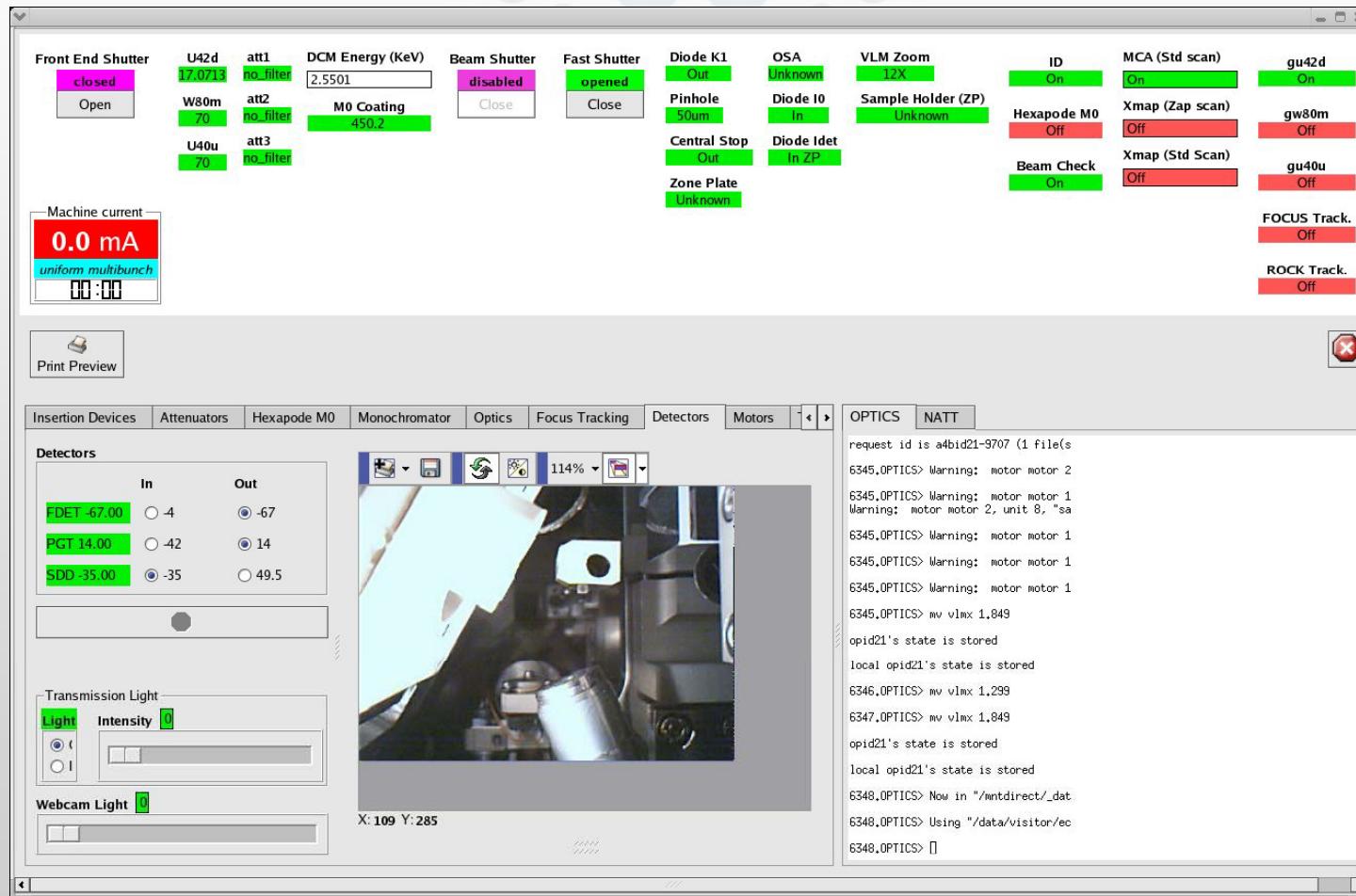
**Oxidis**

# Beamline GUI : BlissFramework

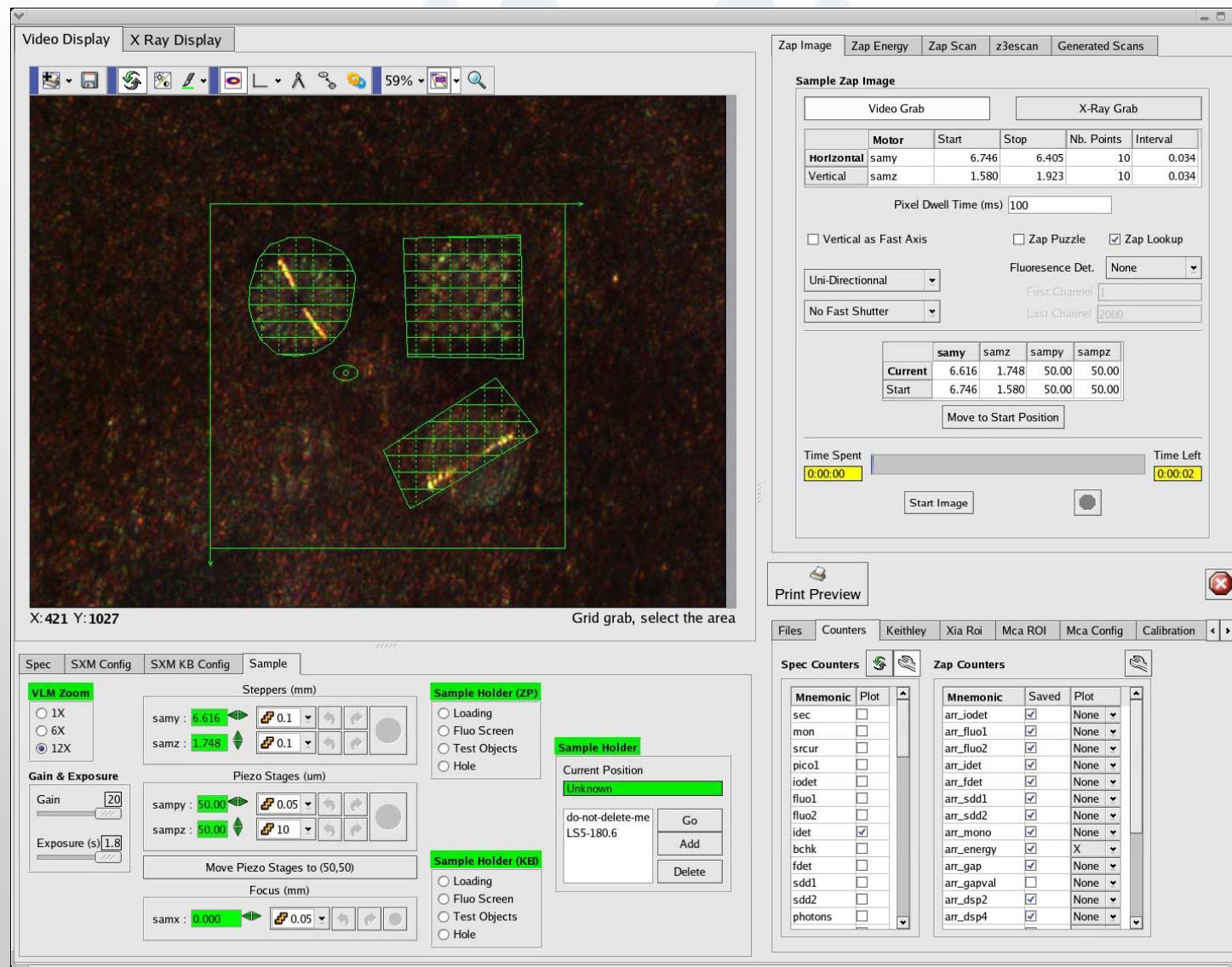
- Written in python/Qt – MVC architecture
- GUI application is made of “cemented” bricks
- Communication to SPEC: **server mode**
  - Socket interface
  - SpecClient python module
- First GUI: MxCube (MaxLab, Soleil, Bessy)



# Beamline GUI : example



# Beamline GUI : example



# Evolution : on-going

- Common 2D detectors framework : **LIMA**
  - More detectors to come
  - New interface in spec
- More beamline GUI:
  - **BlissFramework4** ready
  - First application : BioSaxsCube
- **TANGO** to replace **TACO** whenever possible
- Hardware upgrade : replace VME

# Evolution

- Integrated beamline configuration tool
- Continuous scans
  - Easier integration on beamline
  - Easier integration of detectors
- New data format:
  - Specfiles + EDF Files  $\Rightarrow$  **HDF5**
- Experiment database : ISpyB ??
- keep SPEC ??

# Evolution : alternatives to SPEC

- Evaluation of 3 control systems:
  - GDA (diamond)
    - Most advanced solution
  - SARDANA (alba)
    - Based on already used tools (python, qt, tango)
    - No beamline yet
  - In-house solution

## Conclusion

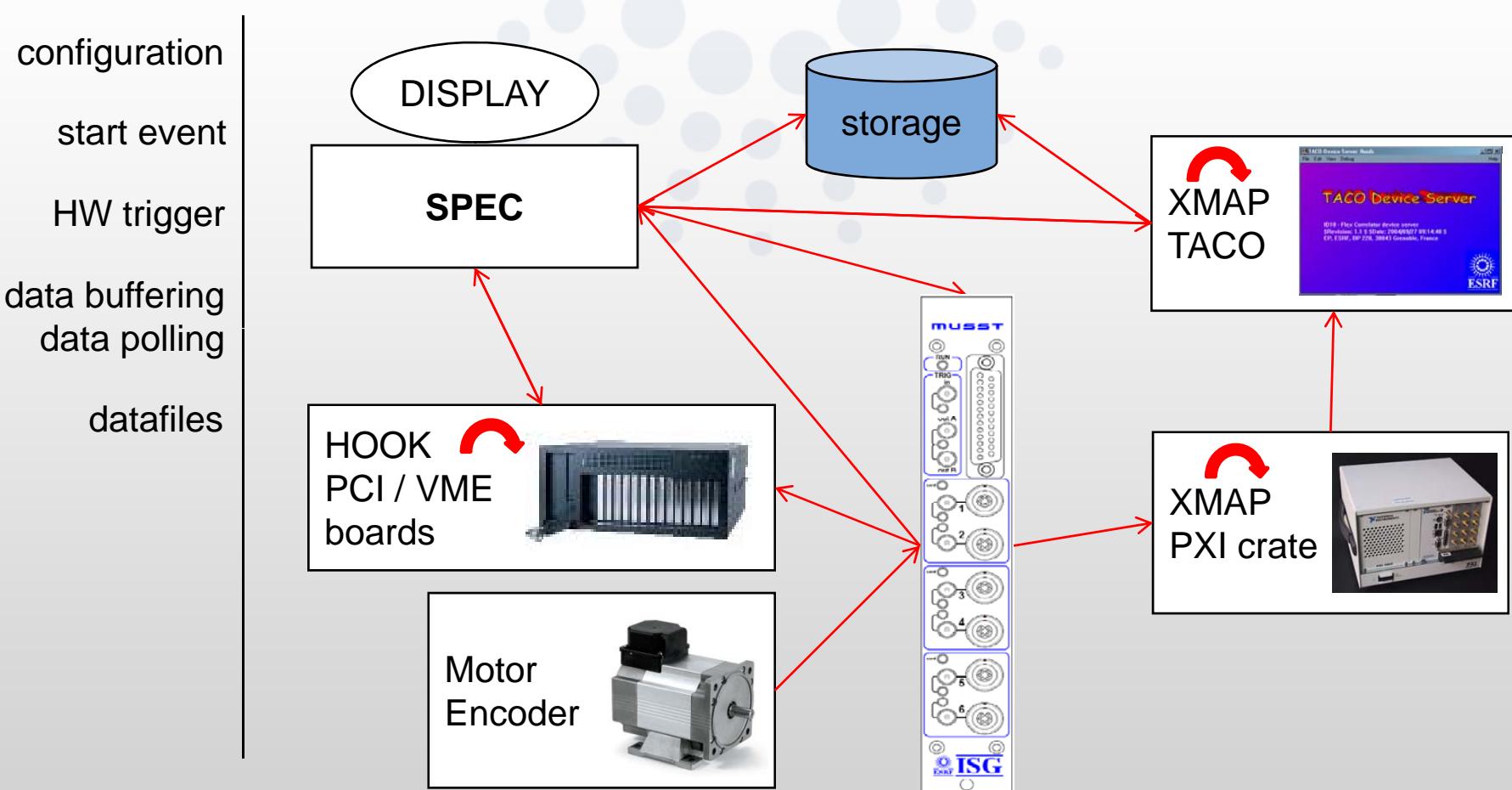
- Keep improving existing system
- Maintain support of running beamlines
- Prepare new upgrade beamlines (start in 2011)
- Investigate new solutions



*Thank you!*



## Continuous scans



# 2D detectors : Camera Link

- **FOCLA module**

- Connects to ESPIA
- 2 X Camera Link connectors
- Multiple pixel packing formats
- Test image generator @ ~180 MB/s



- **Dalsa Pantera 1M60**

- Frame transfer technology
- $1024 \times 1024 @ 60 \text{ fps} \Rightarrow 120 \text{ MB/s}$

- **Sarnoff CAM512**

- $2 \times 8 \text{ ADCs} - 12 \text{ bit}$
- $512 \times 512 @ 300 \text{ fps} \Rightarrow 150 \text{ MB/s}$
- $512 \times 128 @ 500 \text{ fps} \Rightarrow 125 \text{ MB/s}$

